Technical Documentation

for the

Mass Calibration Laboratory

Balance Automation

Written by

Vincent J. Lee

August 1998

Automation Production Technology Division

National Institute of Standards and Technology

NISTIR 6283

Blank page.

Table of Contents

Blank page.

I. Introduction

This report documents the partial automation of four Mettler AT-type, precision balances in the Mass Group's mass calibration laboratory[1]. The automation software was written in the LabVIEW version 4.01 programming environment running in Microsoft Windows NT, version 4.0. The computer, balance and other instruments in the automation set up communicate via RS-232 serial lines.

The Mass Group is officially part of the Acoustics, Mass and Vibration Group of the Automated Production Technology Division in the Manufacturing Engineering Laboratory at NIST. It provides mass calibration services for customers who include state and government laboratories and the private sector. This particular mass laboratory where the automation is employed is used to measure weights with masses of one kilogram or less.

II. Equipment Set Up

The equipment used in the automation includes a barometer, hygrometer, thermometer, two code-operated matrix switches (COMS), laptop computers, and precision balances. These are all connected via RS-232 serial cables through the COMS. The COMS allow the computers to access the instruments when they are needed. One port in the COMS can be connected or "linked" to another that is free or "unlinked". The COMS has a communication protocol that allows any two of its ports to be linked together by the computers using ASCII command strings – thus linking the instruments connected to the ports to each other. Please refer to **Appendix C & D**. In this application, the two COMS are connected together by a port in each. Like the computers and balances, COMS #2 is a serial-device connected to COMS #1. A discussion on the operation of the COMS is included in the next section. Illustrated below are the balances, instruments and their cable connections. Note that the thermometer unit has four probes, each connected to a balance.

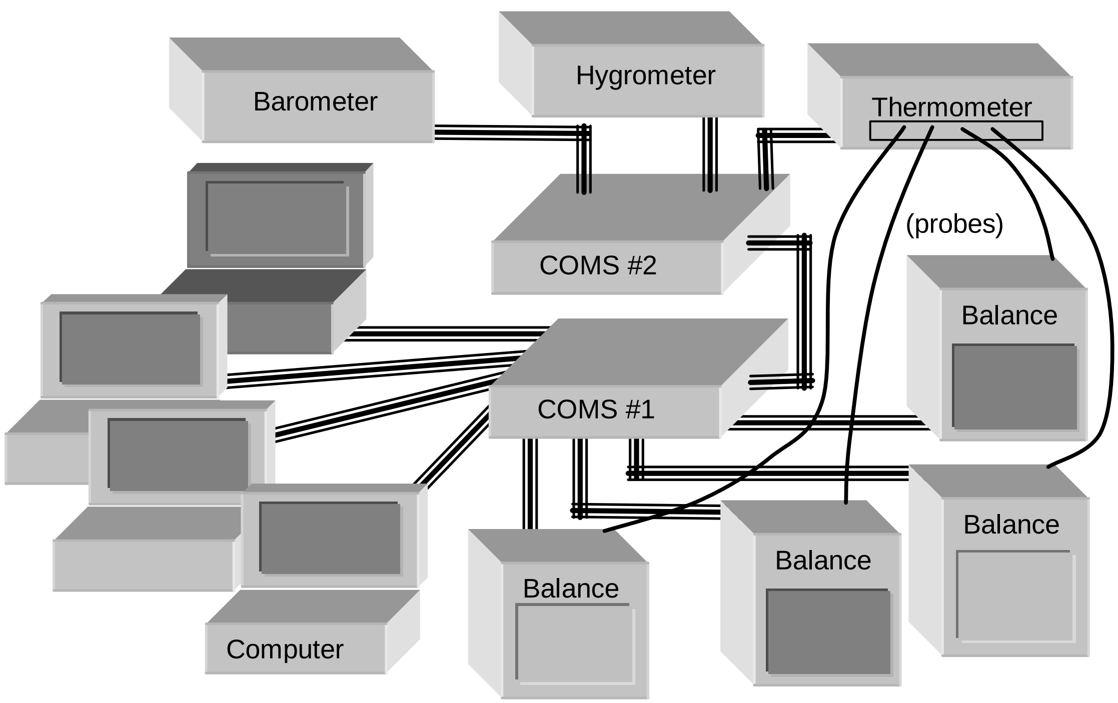

Figure 1. Equipment set up.

[1] Disclaimer. Commercial equipment, instruments, or software are identified in this paper for the purpose of describing the experimental setup. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose or the only equipment that can be used.

III. Automation Software

1. Function and Operation

The balance automation software was written to improve and ease the process of mass calibrations. Previously, mass calibration staff would measure a set of weights and record, manually on paper, the balance readings for each weighing step in the weighing process. This introduces dust into the measuring environment and inaccuracies in data collection. The weighing steps are dictated by the measurement design chosen by the staff according to the weight set. Proper sequencing of the measurements must also be done by hand – most likely for long and tedious designs. Environmental values of air temperature, atmospheric pressure and relative humidity are recorded twice – once before the weight measurements and once after – also manually. Note as well that although the balances are electronic and capable of serial communications with computer equipment, they were used manually.

The automation software's function allows one to four operators, each at any of the four stations, to perform measurements simultaneously. The software coordinates the balance and environmental readings by connecting the computer to the instruments via the COMS boxes for each mass measurement. It communicates with the thermometer, barometer, hygrometer and balance through links between the ports of the COMS boxes to get their readings, records the data values in files on the data disk, and instructs the operator on when to place weights on the balance pan.

After the measurement of the weights is completed, the software filters the data for operator-rejected measurements, combines the data from previous series if they exist, and creates an input file that is formatted for the Mass Code software program. The Mass Code runs (and was developed) separately from this automation software. Note that the input file is created only after a single-series design measurement or after the last series of a design is complete. Also, the data file may be read into a spreadsheet program for custom analysis.

To run the software[2], the operator double clicks on the desktop icon labeled "bal". The front panel appears with instructions for the operator in the large message box in its lower, left-hand quarter. Other features of the front panel include boxes for the temperature, pressure, relative humidity readings, and the balance reading; buttons for editing measurement information, setting the hardware configuration, and displaying weight and design information; and a display of the measurement progress. Also included are menu boxes for choosing the balance and design type.

After the operator starts the software, he/she is instructed to choose which station is used: station #1, #2, #3, or #4. Each station corresponds to a different balance. Next, the operator chooses the measurement design. These include the designs that are commonly used in the mass laboratory. After making these choices, the operator clicks on the button labeled **Edit Info** to enter customer and measurement information.

When editing information, the operator is presented with four sequential screens in which he/she enters the information. Each item in a screen has a field where the information is typed. After all items in a screen are entered, the operator clicks on the button used to continue to the next step. **Appendix A** includes a list of these items and their descriptions.

After all the information is entered, the operator is instructed to click on the **Set Configuration** button. This causes the program to initialize and check the thermometer, barometer, and hygrometer setups. When done, the operator is then instructed to click on the **Start** button to begin the measurement process.

To measure the mass of a weight, the software displays a dialog box. The box lists by name the weight (or weights) that are to be placed on the balance pan. Also the door of the balance is opened automatically with a "beep" sound and an "OPENING…" message on its liquid-crystal display. After the operator puts the weight on the pan he/she clicks on the **PROCEED** button in the dialog box (or presses the space bar or

---

[2] See Lee, Vincent J., "Operator's Manual; Balance Automation Software for the NIST Mass Calibration Laboratory" (NISTIR 6276) for instructions on using the software.

return key on the laptop computer) to indicate that the weights are ready for measurement. The balance door automatically closes with a "beep" and "CLOSING…" message on its display. The software then begins taking readings from the hygrometer, barometer, and thermometer. While these are being read, the balance is stabilizing. A 30-second stabilization period allows enough time for reading the other instruments. The balance is then instructed to send a stable reading that is recorded into a data file. The balance door is re-opened and the dialog box instructing the operator to place the next weight on the pan re-appears. After each measurement the dialog box allows the operator to re-measure the weight before measuring the next weight. A re-measure is done at the discretion of the operator and the rejected measurement is marked in the data file and is later filtered out of the input data for the mass code software. After all the weight measurements are made in a series, the operator has access to the different data files created and stored on a floppy disk. A dialog box displays the name and location of the data file holding all the data as measured as well as the file formatted for the mass code. If all the series of a design have been finished, a file containing all the series is created for the operator to use as input to the mass code software.

2. Algorithm

The software runs a simple algorithm to control the measurement process. First, it sets variables it uses, sets the computer's serial communication port, allows the operator to enter measurement and customer information. Then it runs a loop to read the environmental instruments and balance and records the results.

The variables include the home directory path name, data file names, and various flags and arrays. The values put into these variables include those that are encoded in the program, read from data files, and entered by the operator.

The main loop of the algorithm (as referred to in the figure below) consists of opening the balance door and displaying in a dialog box an instruction to the operator to place the weight or weights on the balance pan. After placement, the operator signals the software to close the door by clicking on the dialog box button. The software then reads the hygrometer, barometer, thermometer, then the balance. The values are then stored in a data file (referred to as the "raw data file"). For each weight, the program then checks if the operator wants to re-measure. If re-measuring, the operator leaves the weight on the pan and the process of reading the instruments is done over again. If the operator is satisfied with the measurement, the program goes on with the next weight.

After measuring all the weights in the series, the program finishes the series by converting the data file containing all the measurements, satisfactory and rejected, into a formatted input file for the Mass Code. The operator is told of the location of the raw data file and Mass Code input file via a dialog box.

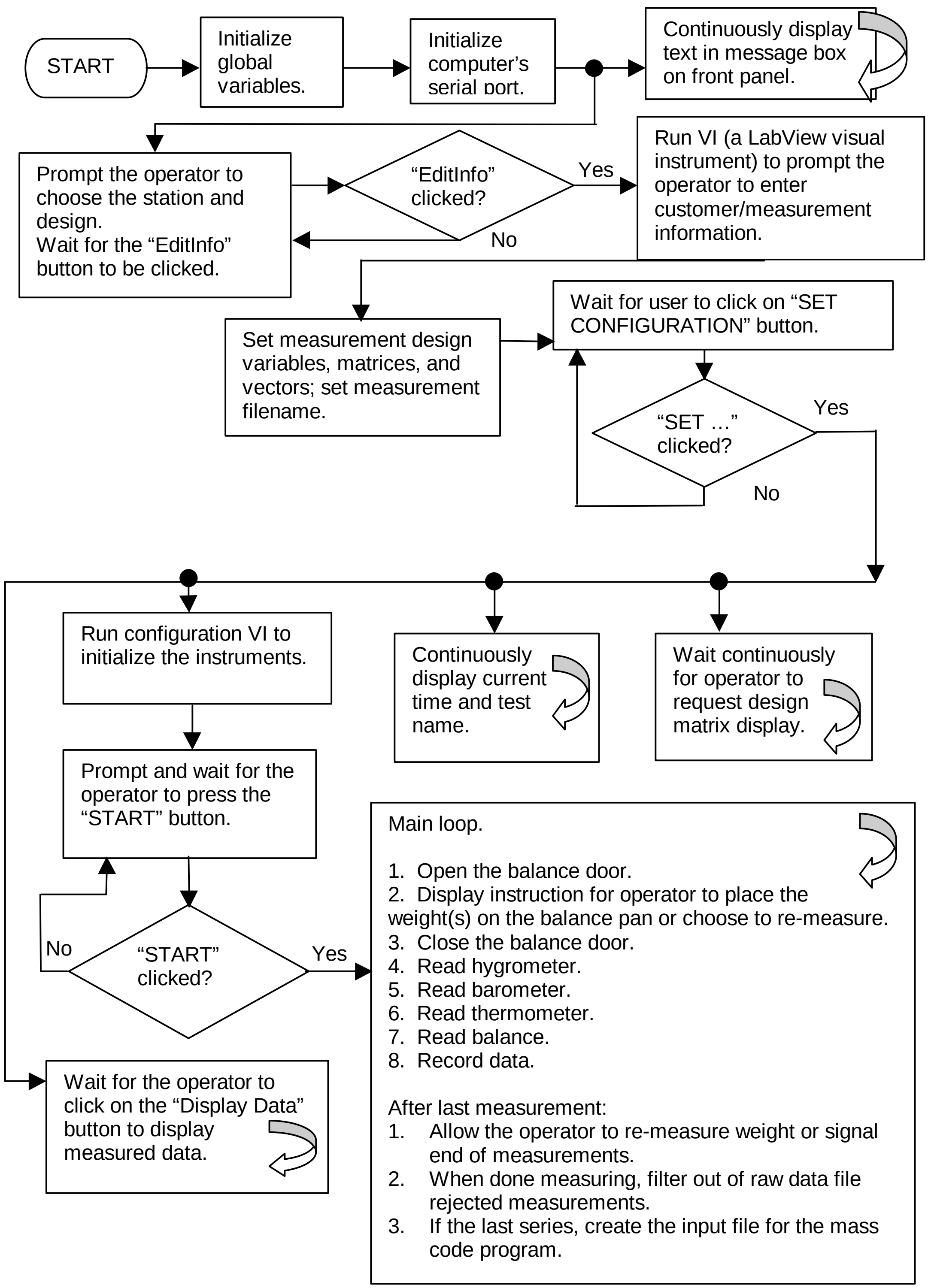

Figure 2. Algorithm flowchart.

3. Controlling the Code-operated Matrix Switches

In order for the software to communicate with the instruments, the COMS must be configured to connect an instrument with the computer's serial communications port. (Refer to **Appendix E**, Figure 14 or **Appendix C**, Figure 8.) In addition to the permanent cable connection between port #0 of each COMS box, the software sends commands to COMS #1 to electronically link the computer's serial communication port to port #0 of COMS #1. Then the software sends a command to COMS #2 via port #0 of COMS #1 to link COMS #2's port #0 to the instrument's port. To connect these or any two ports together with the software, the COMS command, **L**, is sent by the computer in the format dictated by the COMS hardware.

In their communication protocol, each COMS uses an attention character to start a command string. (See **Appendix F**) For COMS #1 it is **%**; for COMS #2 it is the default **@**. For COMS #1 the format of the link command is **%L*xy*** where **%** is the attention character, **L** is the link command, and ***x*** and ***y*** are the numbers of the ports that are linked. Similarly, the command is **@L*xy*** for COMS #2. After the link command is sent, the COMS returns a status string saying whether or not the link was successfully made. Similarly, a command to unlink the ports is used when the computer is done communicating with an instrument. For this the **U** command is used. The commands are **%U*xy*** and **@U*xy*** for COMS #1 and #2, respectively.

4. Development of the Software

The automation software was developed using the LabVIEW® programming environment. This was chosen because it allows the easy development of the human interface and integration of serial devices with a computer.

The software was written in separate modules with each module performing a distinct task. Each module consists of a front panel and a diagram. The front panel contains any controls or displays the programmer may wish to use. Depending on the purpose of the module, the front panel may or may not be seen by the program user (e.g., global variables do not have front panels). The diagram contains the graphical program code for the module and is not seen by the program user. A module is called a virtual instrument or VI.

5. Software Organization

The software's VIs are hierarchically organized. What's referred to as the top-level VI contains a front panel that acts as the main operator interface of the software. This VI contains the code that controls the measurement process. From here, other sub-VIs are called to perform their tasks. These may also include a front panel for the operator to enter or access information about the measurement.

6. VI Descriptions

The following table contains the names of the VI's in the software and their descriptions. All these VI's have the prefix "SerialVJL" and suffix ".vi" in their names. These are stored in a library called Balance.llb.

Table 1 . VI Descriptions.

Main	The VI that contains the front panel operator interface and the code to control the overall measurement process.
ArrayRawData	Stores the measured data in the variable "Global Array Raw Data". These data include the time stamp, temperature, pressure, relative humidity, and balance reading and measurement status (saved or rejected).
CheckCOMS1Links	Used to retrieve the current links made in code-operated matrix switch #1. The links are retrieved by sending the **%Q** command to the COMS. The status report string is returned and parsed for the desired linkage.
ClearSerialPort	Re-initializes the serial communications port on the computer.
CmdStatLog	Logs the command and status messages it receives in a file.
ConsolidateFinalReport2	Creates a report file for the just-measured series. This file is located in the c:\autobal directory and has the extension ".rpt". Each series measured has a report created for it. Each report's contents is combined with the weight description file (extension ".wgt") and the measurement data file (extension ".dt3").
ConsolidateFinalReportsForMassCode	Creates an input data file that is used by the Mass Code. This file has the extension ".ipt" with the name of the first measurement series. If there is only one series in the measurement then the ".rpt" file is simply copied into the ".ipt" file. If there are more than one series, then parts of the ".rpt" files are appended to each other in the order of measurement.
ConvertRawDataFile	When a series is executed, the measurement data are saved in a file with the extension ".dat". This file contains all measurements whether they be wanted or rejected data. This VI removes the rejected data from the file and saves the filtered data in a file with the extension ".dt2".
DataTable	Displays the values of the measurements in a series.
DecipherMassStatus	The value is parsed from the balance status message in double floating-point form before it is saved in the data file.
DisplayDesignMatrix	Displays the design matrix for the operator's information purposes in a window. It also includes the weights' descriptions.
DoorClose	Signals the balance to close its door with an accompanying beep and message display.
DoorOpen	Signals the balance to open its door with an accompanying beep and message display.
DoorSelect	Allows the operator to select the balance's door operation mode. This is not currently being used. Its control is hidden and code is disabled. This VI remains for optional future use.

EditMoreInfo2	Contains a window to allow the operator to enter information about the measurement. This information includes: type B uncertainty in the restraint (mg), nominal temperature (°C), starting restraint ID number, standard deviation of balance (mg), sensitivity value used, multiplier for computed A's[3], accepted between standard deviation of the process (mg), and the check standard number. After retrieving these values, the VI creates lines 8, 9, and 10 of the mass code input data file format and puts them into the ".rpt" file. Line 13 is also created here but is put into the ".wgt" file.
EditVectors4	Prompts the operator to enter the restraint vector, check standard vector, restraint vector of the next series (if applicable). This information is then stored in the ".wgt" file.
EditWeightNames	Prompts the operator for the names and attributes of the weights being measured in the series. These are stored in the ".wgt" file.
FindFilenameMatch	Used to check for a used file name after the operator enters a name for the series that is about to be measured.
GetDesignMatrix	Retrieves the design matrix, given the design name, from the appropriate file with and ".mtx" extension. This data is stored in memory in the global array called "design matrix".
GetDesignVectors	Retrieves the design vectors from the appropriate ".vct" file. These vectors control the measurement sequence of the weights. These vectors are stored in memory in the global array "Global Design Vectors".
GetFileSize	Gives the file size in bytes, given the pathname.
GetInitFinEnviroData2	Determines the first and last values of the temperature, pressure and humidity of the series (along with their corrections) and stores them in the file with extension ".env". These values are read from the ".dt2" and c:\autobal\system\correct.dat files.
GetSeriesICustInfo	Prompts the operator to enter the customer's name and address, measurement description, and test number. These are the values of lines 1 to 7 in the mass code input file format. The date is automatically entered. These values are stored in the ".rpt" file.
GetThermLinkSemaphore	Retrieves a semaphore, represented by a global Boolean variable, "GlobalThermSemaphore", so that the thermometer may be accessed without colliding with other computers' requests. If the semaphore is "true" then someone else has access to the thermometer; if it is "false" the thermometer is accessible. This VI was created to control multiple accesses to the thermometer by the same station in possible future versions.
GiveThermLinkSemaphore	Releases the semaphore by setting the "GlobalThermSemaphore" to "false".

[3] The "A's" are the average balance differences for double-substitution type weighing. Please refer to NBS Technical Note 1127, "National Bureau of Standards Mass Calibration Computer Software", R. N. Varner & R. C. Raybold, July 1980.

InitBarometer	Initializes the barometer before the measurements are made during the "Set Configuration" step. This performs the memory check, stack check and sets the baud rate and parity of the barometer. If an error is encountered the operator is notified by a display window and instructed on what to do.
InitDataTable	Initializes the "Global Array Raw Data" variable with zero values.
InitRelHumidity	Initializes the hygrometer by sending its stop command.
InitThermometer	Initializes the thermometer by setting the units to Celsius and choosing three decimal places.
LinkBalance2	Links the computer to the balance via the COMS box. It first sends the command to make a link in COMS #1 from the port where the computer is linked to the port where the balance is linked.
LinkBarometer	Links the computer to the barometer. It first sends the command to make a link in COMS #1 from the port where the computer is linked to port 0. Then a command is sent to COMS #2 to link its port 0 to the port where the barometer is linked.
LinkRelHumidity3	Links the computer to the hygrometer. A command is sent to COMS #1 to link the computer's port to port 0. Then a command is sent to COMS #2 to link its port 0 to the port where the hygrometer is linked.
LinkThermometer3	Links the computer to the thermometer. A command is sent to COMS #1 to link the computer's port to port 0. Then a command is sent to COMS #2 to link its port 0 to the port where the thermometer is linked.
ParseCOMSLinks	Parses the status message that was returned as a result of the **`%Q`** command sent to COMS #1. This looks for a link between two specified port numbers in the message and returns a result of "TRUE" (if a link was found) or "FALSE" (if no link found).
ReadHumid3	Sends a command for the hygrometer to return the current relative humidity.
ReadMass	Sends a command to the balance to return the current balance reading.
ReadPress	Sends a command to the barometer to return the current pressure reading.
ReadTemp3	Sends a command to the thermometer to return the current temperature reading.
RecordRawData	Records the just-read temperature, pressure, humidity and balance reading into the raw data file (extension ".dat"). This is done after each measurement.
ReformatDate	Reformats the date given by the canned date & time VI to the format required by the mass code program.
ReformatRawDataFile2	Converts the ".dt2" data file into the ".dt3" file. This new file contains the calculated average values derived from the raw data using the (a – b – c + d) / 2 formula and appends the `–200000` string at the end.

SerialCfg	Configures the serial communication port of the computer for the baud rate, data bits, stop bits, parity, and sets the COMS port numbers for the thermometer, barometer, hygrometer and balance. These values are retrieved from the configuration file, c:\autobal\system\config.dat.
SerialCfgCommPortOnly	Configures only the serial communication port of the computer, i.e., baud rate, data bits, stop bits, and parity.
SerWriteRead	Sends a command to a device then reads a status message from the same device.
SerWriteRead2	Based on SerWriteRead, sends a command to a device then reads the status returned one byte at a time until a carriage return character is encountered.
SerWriteRead3	Based on SerWriteRead2, sends a command to a device then reads the status returned one byte at a time and when done also returns the length of the status.
UnitSelect	Allows the operator to choose the units used by the balance (grams or milligrams) during the initialization of the balance. The default is milligrams. This is not currently done but exists for future use.
Unlinkbalance	Unlinks the balance from the computer. The link between the balance and port 0 of COMS #2 is first deleted then the link between port 0 and the computer in COMS #1 is deleted.
UnlinkBarometer	Unlinks the barometer from the computer. To do this, the link between the barometer and port 0 of COMS #2 is undone followed by the link between port 0 and the computer in COMS #1.
UnlinkRelHumidity2	Unlinks the hygrometer from the computer. The link between port 0 and the hygrometer in COMS #2 is first deleted then the link between port 0 and the computer is deleted.
UnlinkThermometer	Unlinks the thermometer from the computer. This is done by unlinking the thermometer from port 0 in COMS #2 followed by unlinking port 0 in COMS #1 from the computer.
VibrationsSel	Selects the vibration mode for the balance as chosen by the operator. This, however, is not currently done but remains in the code for possible future use.
WeighProcSel	Sets the balance's weighing process according to the operator's choice. This is not done but its code still exists for possible future use.
WhichWeight	Returns the name of a weight, in string form, for display in a dialog box instructing the operator on which weights to put on the balance pan. This uses the "Global Design Vectors" variable and the "Weight Names" array to determine the weights.

7. Data Files

There are basically four data files read by the software when the program is run. The first is the configuration file, c:\autobal\system\config.dat. It contains the computer's serial communication port settings as well as the port numbers of the instruments at the code-operated matrix switches. The second file contains the corrections for the thermometer, barometer, and hygrometer. The file is located at c:\autobal\system\correct.dat. The config.dat file is not edited in normal use. A change in this file should correspond to a change in the hardware configuration. The correct.dat file, however, is edited periodically. The changes made here correspond to new corrections resulting from the re-calibration of the instruments. Refer to **Appendix B** for detailed information on these files.

The remaining two types of files are the design matrix and design vector files. Derived from some of the measurement designs outlined in NBS Technical Note 952 (see footnote in **Appendix G**), they have the extension ".mtx" and ".vct", respectively. An ".mtx" file contains the measurement design and is used for display purposes for the operator to see. It displays the design with "+", "–", and "0" characters. The measurement sequence is regulated by the ".vct" file. It determines which weight is measured and when and uses the same characters. Note that in these files each place in a line that these characters occur represents a weight in the weight set. A "+" or "-" means that the weight is measured and multiplied by a +1 or –1, respectively. A "0" means that the weight is not measured. The operator does not see this file but its result. Each design has a design matrix file and design vector file located in the directory c:\autobal\system. See **Appendix G** for a listing of these files. Also see **Appendix H & I** for the contents of these files.

There are seven files that are created by the software each time the program is run. These are stored in a separate directory on the floppy drive, A:\, to allow the operator to move his/her data from station to station in order to connect related series. Three of these files record the measurements made. The first contains the measurements as they are taken. This file has the extension ".dat". This file contains in each row the measurement number (starting with "1"), a time stamp, the temperature, the pressure, the humidity, the balance reading, the name or names of the weights measured, and the result of the measurement – whether it was saved by the operator or rejected. The next file created, with the extension ".dt2", is a reformatted version of the ".dat" file with any rejected measurements removed. The third file, with extension ".dt3", contains a reformatted version of the ".dt2" file containing the average values of the measurements (the "A's") in the format required by the Mass Code input file's line 21. (Line numbers are used to refer to the input file items. Some of these items are from the operator and others are derived from the measurements taken.) It is the ".dt3" file that is used in creating the series' report.

The fourth file is created after the measurements of the series are taken. This file has an ".env" extension and contains the first and last environmental measurements (temperature, pressure, and relative humidity) as well as the corrections for the instruments measuring these values. This file is line 11 of the Mass Code input file. This file is first appended to the report file.

The fifth file is the weight attribute file. It contains the names and attributes of the weights measured within a series and has the extension ".wgt". This file is also included in the report file and contains lines 14 through 20 of the Mass Code input file.

The sixth file, the report file, is the one that is created in the same format as the Mass Code input file. It contains all the lines of the mass code input file including the line identifying the end of the series and the end-of-data terminator (`STOP`). If this series is the only one for the set of weights, the ".rpt" file is simply copied into the Mass Code input file. If this series is one of several for the set of weights, then it is disassembled and appended to the first series' ".rpt" file to make one Mass Code input file.

Finally, the last file created by the software is the Mass Code input file. This is created by appending the report files (".rpt") created for a set of series together or, for a stand-alone series, simply copying the report file to the input file. This input file has the extension ".ipt".

8. The Mass Code Input File and the Automation Software

The mass code input file is an ASCII-text file created after the measurements are made. The input file has 23 or more lines depending on the number of series included in the measurement. For all input files its first seven lines contain an identification of the measurement including the customer name, 2-line customer address, 2-line description, date and test number. The operator provides these items to the "EditInfo" VI screen. The next (eighth) line contains the type A uncertainty in the restraint, type B uncertainty in the restraint, nominal temperature, and starting restraint ID number and is dealt with by the "EditMoreInfo2" VI. Note that the value of the type A uncertainty is automatically set to "0" (zero) by the software. Similarly, the nominal temperature is by default "20" and is also entered in the line by the software.

Lines nine to 22 are repeated for each series in the measurement. Line nine includes the type of weighing (defaults to "4", direct reading, for this automation), type of balance (1-pan default), unit of grams, and "regular" balance readout. Line ten contains the month, date, year, operator number, balance number, and check standard number. Line nine is automatically formatted into the file and line ten is prompted for from the operator. These two lines are created in the "EditMoreInfo2" VI.

Line 11is created after the measurements in the series are completed. This line includes the initial and final temperatures, pressures, and humidity as well as corrections to the initial and final temperatures, pressures, and humidity. These are derived by the "GetInitFinEnviroData2" VI and are initially put into a file (containing only these) with the ".env" extension. They are then put into line eleven by the VI, "ConsolidateFinalReport2".

Lines 12 and 13 are created in the VI "EditMoreInfo2". Line 12 contains the number of observations, the number of weights, the calibration identification number, and the number of linear combinations. Line 13 contains the following operator-supplied items:

- the standard deviation of the balance in milligrams,
- the mass of the sensitivity weight in milligrams,
- the volume of the sensitivity weight in cubic centimeters,
- the volumetric coefficient of thermal expansion of the sensitivity weight per °C,
- the accepted between standard deviation of the process in milligrams,
- and the multiplier for the computed averages.

These values in line 13 are for direct-reading weighing.

The "EditWeightNames" VI creates line 14. This contains a 15-character identification for the weight as well as:

- the nominal value of weight in grams or pounds,
- the density of the weight in gm/cm^3,
- the volumetric coefficient of thermal expansion of the weight per °C,
- and the accepted mass correction in milligrams of the weight if it is a standard.

These items are retrieved from the operator at the VI's screen.

Lines 15 through 19 are created in the VI "EditVectors4". These contain vectors that:

- identify items in the restraint,
- identify items in the check standard,
- identify the restraint items in the next series,
- identify the items to be reported, and
- describe the design matrix for the series.

Values for the lines 15, 17, and 18 may be "0" or "1" and for lines 16 and 19, "1", "0", or "-1" and are also supplied by the operator.

Line 20 contains one or more operator-supplied vectors identifying the linear combinations with "1"s and "0"s. It is optional.

The “ReformatRawDataFile2” VI creates lines 21 and 22. Line 21 contains the balance observations in direct-reading format. This includes the average readings calculated with the formula (a-b-c+d)/2 in a format like:

```
82.2  10082.2
80.1
28.0
63.2
36.2
54.6
18.9
-28.5  9971.5
```

where the first and last lines also contain the sensitivity value used. Line 22 identifies the end of the series with the string `-200000`.

After all series are included in the report each with data in the format of lines 9 to 22, the string `STOP` is appended. This terminates the end of the data with a carriage return following.

9. Conclusion

A number of improvements have resulted from the implementation of the automation setup in the mass calibration laboratory. Because mass calibrations are affected by dust, an environment that is as clean as possible is recommended. The elimination of the paper used to record measurements (and resultant dust) is very helpful in the improvement of the measurement process.

The automation software also records more data than the manual method. For each weight measurement all of the environmental variables are recorded. A clearer picture of how these variables change during the measurement process can be seen than with only before-and-after recordings.

A measurement bias caused by the idiosyncrasies of each staff member’s measurement technique is also reduced. Although they still place the weights on the balance by hand, the automation software determines when the reading is taken from the balance.

The recording of measurement data is also more accurate. Because they are automatically taken and recorded by the computer, readings are correctly recorded in a computer data file. No misread readings are recorded; no readings are mistakenly recorded.

The compilation of the measurement data is automatic and thus more accurate and faster than before. Human error is not introduced into the processing and formatting of the data as required by the Mass Code. Also since the data are recorded in spreadsheet format, they may also be imported into a spreadsheet program and analyzed.

Improvements to the automation will be introduced in the future. Other computer-compatible electronic balances will be integrated into the automation setup as needed; other measurement designs will be included as requested. With ongoing usage, the automation software and setup will continue to be debugged and altered as required by the mass calibration services.

Appendix A. Operator Input Items

This appendix lists the items required from the operator to identify and describe the weights and their measurement.

On the front panel the operator uses the controls to choose the station number and design type.

When instructed the operator clicks on the **Edit Info** button to enter more information. A new window (Figure 4) appears prompting the operator for the:

- customer name,
- customer address (in two lines),
- description of the measurement (in two lines),
- balance number,
- operator number,
- test number, and
- name of the report.

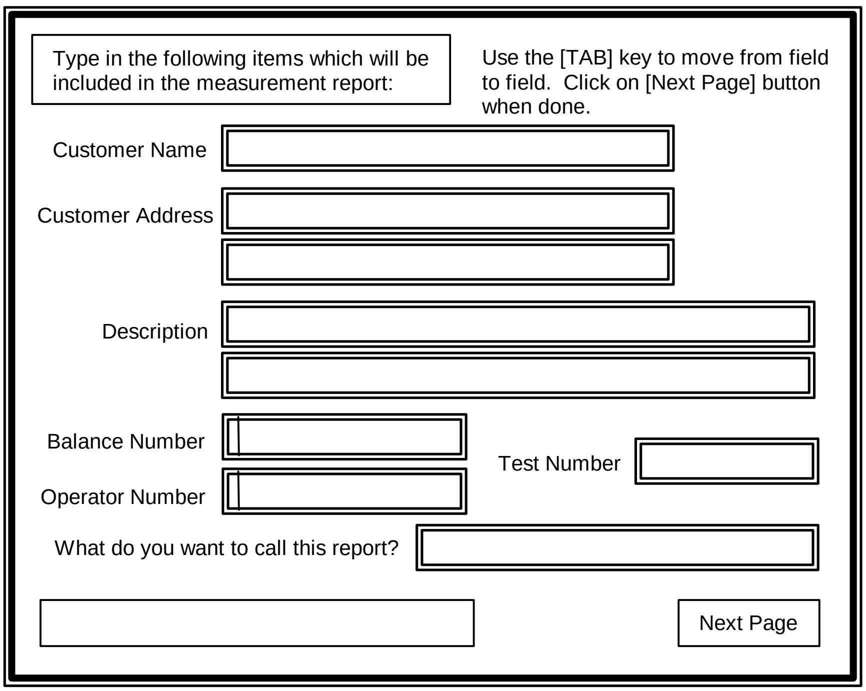

Figure 3. Customer information input screen.

The operator enters these items by typing them in. When done, the operator clicks on the **Next Page** button with the mouse.

The window in Figure 5 appears with prompts for information about the measurement. These include the:

- type B uncertainty in the restraint,
- nominal temperature in Celsius (by default 20, automatically entered),
- starting restraint ID number,
- standard deviation of the balance in milligrams
- sensitivity value used,
- accepted between standard deviation of the process in milligrams,
- multiplier for the computed averages, A’s, and
- check-standard number.

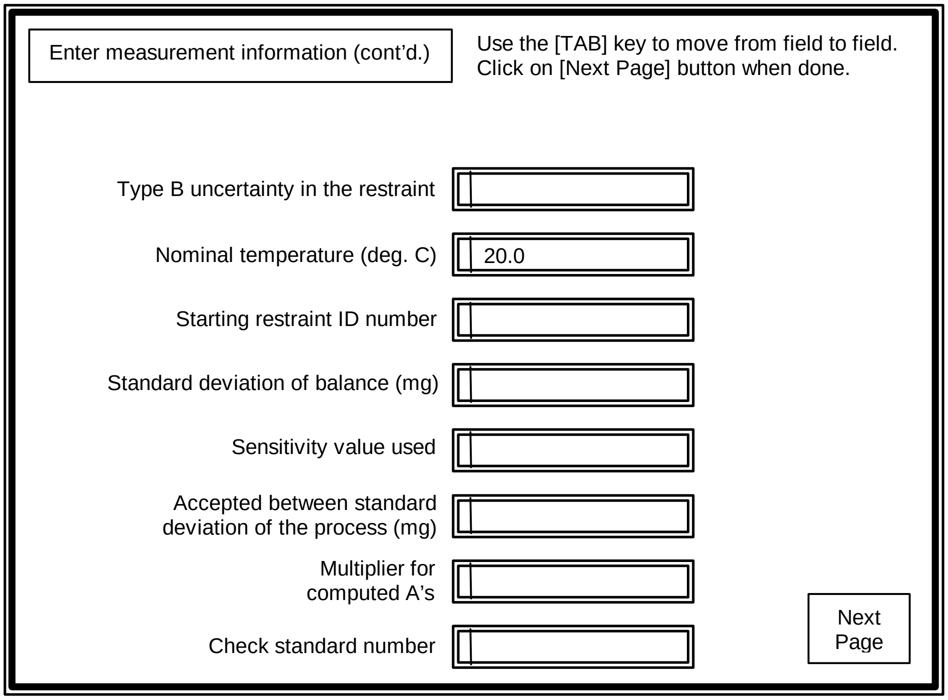

Figure 4. Measurement information screen.

The operator then clicks on the **Next Page** button to continue.

The next window (Figure 6) that appears prompts the operator for information on the weights that will be measured. The operator enters into each field of the window:

- a 15-character name of the weight (or weights),
- the nominal value of the weight(s) (g),
- the weight density (g/cm^3),
- the volumetric coefficient of thermal expansion per ºC,
- the accepted mass correction of the weight(s) (mg).

If the weight is a reference standard weight, the operator checks the box indicating so. This window appears for each of the weights in the measurement in the order of the first weight measured to the last as dictated by the measurement design. After each weight the operator clicks on the **Go to next weight** button.

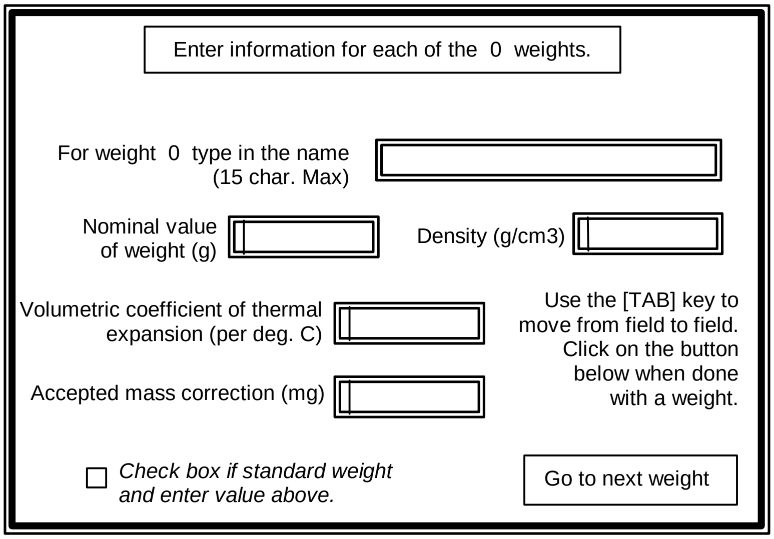

Figure 5. Weight information screen.

When all the weights are done the last window appears.

This last window (Figure 7) prompts the operator to enter vectors for the measurement. These vectors are used to describe the restraint, check standard, restraint items in the next series (if there is no following series this is left blank and the corresponding box is checked) and the report items. These vectors are a combination of zeros and ones delimited by single spaces. When done the operator clicks on the **ENTER** button, the window disappears and the front panel returns.

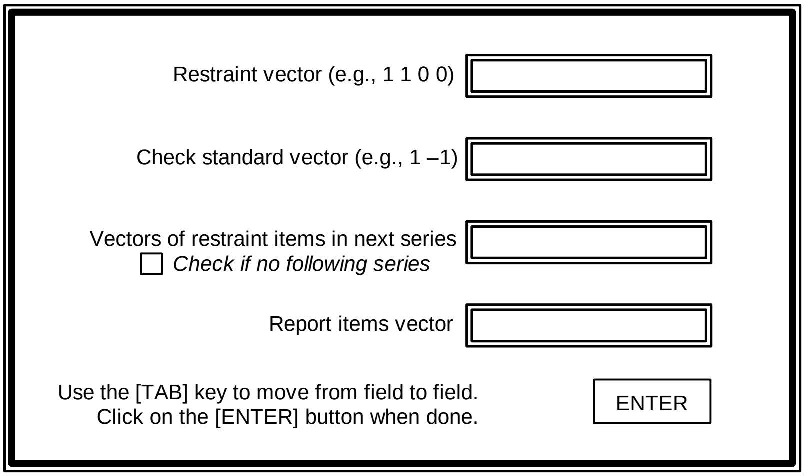

Figure 6. Vector input screen.

Appendix B. Software Data Files

Data files used by the software include configuration files describing hardware set up, measuring designs.

"config.dat" configuration data file:

Contains the serial communication parameters as well as the code-operated matrix switch (COMS) ports for the instruments (thermometer, barometer, hygrometer, balance and computer) at the two COMS units.

The format of the file is:

```
<serial communication port number (on the computer, 0 = COM1, 1 = COM2, etc.)>
<baud rate>
<data bits>
<stop bits>
<parity>
<thermometer COMS port #>
<barometer COMS port #>
<hygrometer COMS port #>
<balance COMS port #>
<computer COMS port #>
<IP address>
```

A sample is:

```
0
9600
8
1
0
4
2
3
5
1
123.456.789.000
```

"correct.dat" instrument correction data file:

This contains the corrections for the thermometer, barometer and hygrometer. It has the format:

```
BAROM, <barometer serial number>, <correction>
HYGRO, <hygrometer serial number>, <correction>
THERM, <thermometer serial number>, <correction>
```

A sample is:

```
BAROM, R3410008    , 0.0001
HYGRO, 64318       , 0.0002
THERM, 1354 003 870, 0.0003
```

"*.mtx" design matrix file:

This contains the measurement designs described by plus and minus signs and zeros. This is used for the display of the design to the operator. As an example, the *.mtx file for the 31S design is as follows:

```
+,-,0
+,0,-
0,+,-
```

Note that only a carriage return character follows the end of the matrix and no spaces follow the commas. No empty lines can follow the text or else the mass code input file will not be correctly formatted. These matrix files are located in the c:\autobal\system directory. See **Appendix H** for a complete listing of these *.mtx files.

"*.vct" measurement sequence file:

This contains the vectors that designate the measurement sequence. The software uses this to control which weights are measured in a series. As an example, the *.vct file for the 31S design is as follows:

```
+,0,0
0,-,0
0,-,0
+,0,0
+,0,0
0,0,-
0,0,-
+,0,0
0,+,0
0,0,-
0,0,-
0,+,0
```

Note that only a carriage return character may follow the vector list. No blank lines can follow the vectors or else the mass code input file will not be correctly formatted. These vector files are located in the c:\autobal\system directory. See **Appendix I** for a complete listing of these *.vct files.

"*.rpt" data file:

This file contains the processed measurement data in a format ready for the mass code software. One of these files is created for each series measured. If only one series exists for the weight set measured then this is the same as the input file for the mass code. Otherwise, there are as many "*.rpt" files as there are series. When all the series are complete these files are combined in the sequence of measurement in the mass code input file with extension "ipt" as described later.

"*.ipt" data file:

This file is created after all the series for a set of weights have been completed. Derived from the *.rpt file it contains one or more series' data in the format for the mass code program. If there is only one series then it is the same as the corresponding *.rpt file. If more than one series exists, it is created from parts of the corresponding *.rpt files.

".dat" data file:

One of these files is created for each series measured. This is the first file containing the measurement data: date and time stamp, air temperature, atmospheric pressure, relative humidity, and balance reading, as well as the weights' names (each are 15 characters long enclosed in double quotation marks) and measurement status (S = saved, R = rejected). Commas separate these items. A sample of lines in the file looks like:

```
1, 9/3/97, 1:04:19 PM, 21.870, 749.7822, 48.00, 0.53000, “wgt_1         ”, S
2, 9/3/97, 1:05:04 PM, 21.860, 749.7651, 48.20, 0.56000, “wgt_2         ”, S
3, 9/3/97, 1:05:45 PM, 21.880, 749.7612, 48.30, 0.56000, “wgt_2         ”, S
4, 9/3/97, 1:06:27 PM, 21.890, 749.7714, 48.30, 0.56000, “wgt_1         ”, S
5, 9/3/97, 1:07:07 PM, 21.890, 749.7714, 48.30, 0.56000, “wgt_1         ”, S
6, 9/3/97, 1:07:45 PM, 21.870, 749.7480, 48.21, 0.56000, “wgt_3         ”, R
6, 9/3/97, 1:08:37 PM, 21.940, 749.7440, 48.00, 0.58000, “wgt_3         ”, S
7, 9/3/97, 1:09:25 PM, 21.930, 749.7339, 47.80, 0.58000, “wgt_3         ”, S
8, 9/3/97, 1:10:07 PM, 21.900, 749.7347, 47.80, 0.57000, “wgt_1         ”, S
```

“.dt2” data file:

This data file is the same as the “.dat” file except that it does not contain the rejected measurements denoted by the R at the end of a line. A VI in the program filters out these lines.

“.dt3” data file:

This data file is derived from the “.dt2” file and is formatted as line 21 for direct reads in the mass code input file. When creating the *.rpt file this is copied into it. A sample looks like:

```
-0.015000   9999.98500
-0.015000
-0.025000
0.000000
0.01000
-0.00500
-0.00500
0.010000
-0.05000   9999.995000
```

Appendix C. Hardware Set Up

This section includes a schematic of the hardware set up.

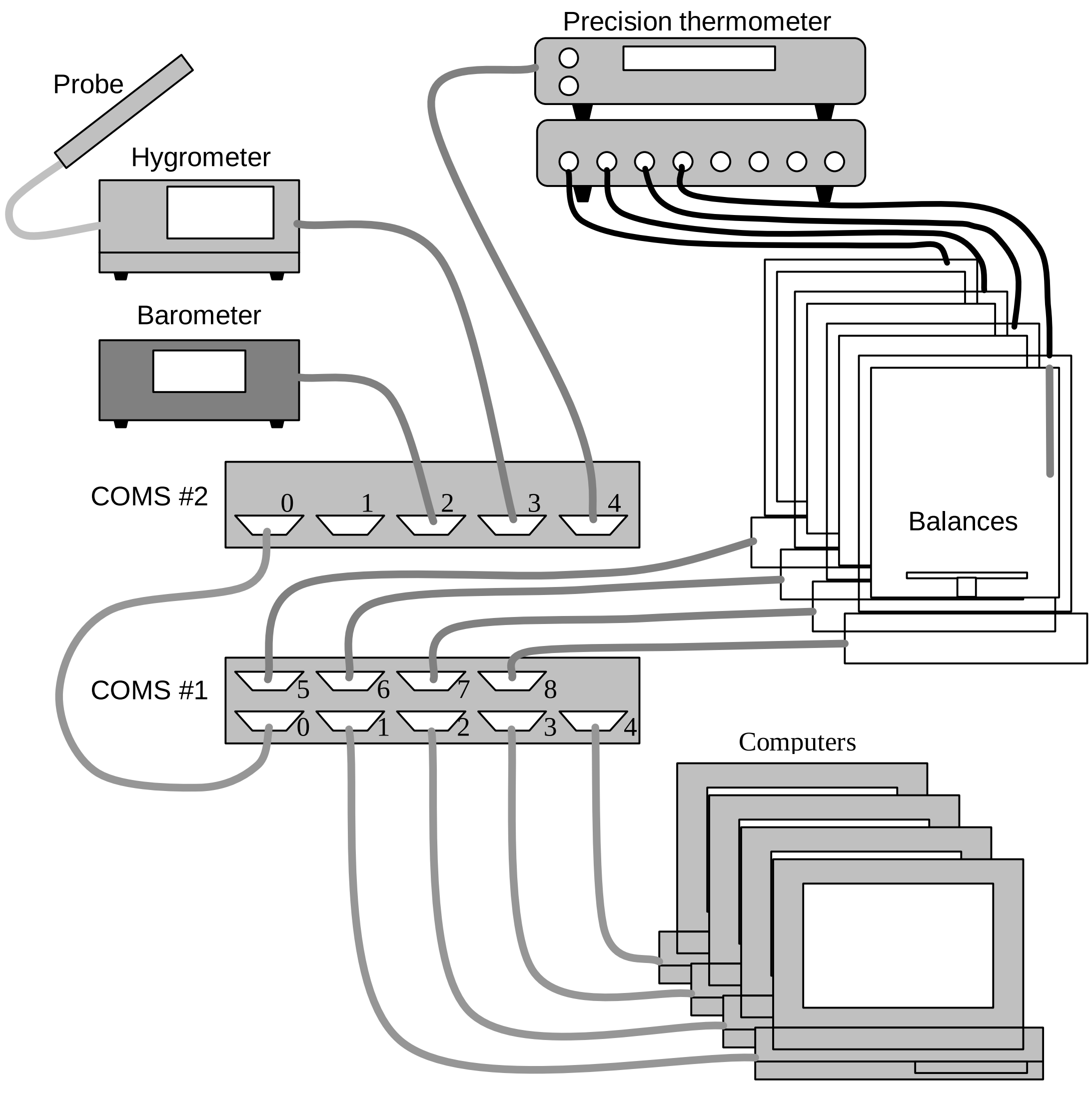

Figure 7. Illustration of hardware set up.

Appendix D. Cabling Pin Assignments

A description of the data cables connected to each device in the automation follows. They are organized by device.

AT-type balance data cables

The balance cable consists of 2 cables: one Mettler-210492, one 5-conductor 25-pin M-F. Each balance data cable is made of two separate cables connected in series: one 5-conductor shielded RS-232 cable plus one data cable from the balance manufacturer. The shielded cable is connected at one end to a port on the COMS box and at the data cable at the other. The data cable is connected to the balance's interface box.

baud = 2400
data bits = 7
parity = even
stop bits = 1

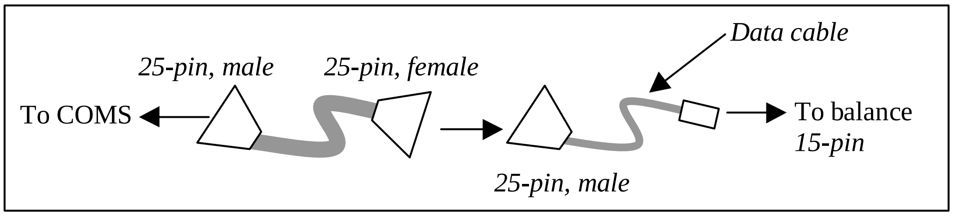

Figure 8. Balance data cable assembly.

5-conductor, RS-232, 25-pin D-subminiature connector pin assignments at COMS #1 (straight-through):

color	*pin number*
Green with white stripes	2
Blue with white stripes	3
Orange with white stripes	4
White with green stripes	5
White with blue stripes	7

Mettler 210492 data cable: (connects above cable to balance)

Color	*15-pin connector*	*25-pin connector*
Green	2	2
Blue	12	3
Orange	3	4
White/green	4	5
White/blue	13	7

Barometer: Paroscientific, Inc., model 740-16B, Digiquartz ® Digital Barometer

baud = 1200
data bits = 8
parity = none
stop bits = 1

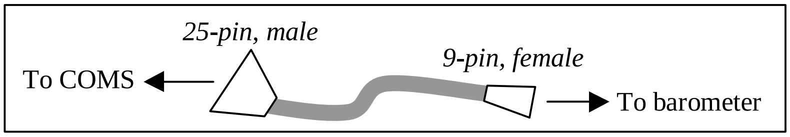

Figure 9. Barometer data cable assembly.

RS-232 25-pin D-subminiature connector pin assignments at COMS #2:

Color	*25-pin connector*	*9-pin connector*
Black	8	1
Brown	3	2
red	2	3
Orange	20	4
Yellow	7	5
Green	6	6
blue	4	7
Violet	5	8
gray	22	9

Thermometer: Automatic Systems Laboratories, Inc., Model F250, Precision Thermometer

baud = 9600
data bits = 8
parity = none
stop bits = 1

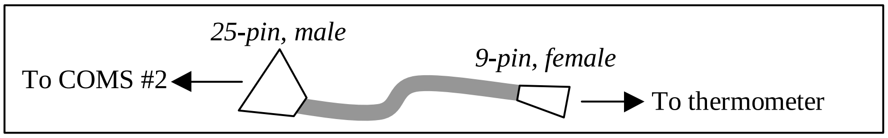

Figure 10. Thermometer data cable assembly.

Color	*25-pin connector*	*9-pin connector*
brown/white	2	3
white/orange	3	2
blue/white	4	7
white/blue	5	8
white/green	6	6
green/white	7	5
orange/white	8	1
white/brown	20	4
grey/white	22	9

Hygrometer: Vaisala, Model HMI38-U064en-1.3, Humidity Data Processor

baud = 2400
data bits = 8
parity = none
stop bits = 2

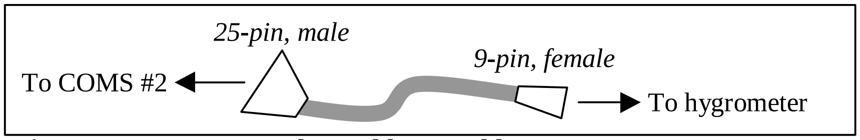

Figure 11. Hygrometer data cable assembly.

RS-232 25-pin D-subminiature connector pin assignments at COMS #2:

color	*pin number*
brown	2
red	3
yellow	7
green	5
green	6
green	8
green	20

Note: green wires are shorted.

Laptop computer:

baud = 9600
data bits = 8
parity = none
stop bits = 1

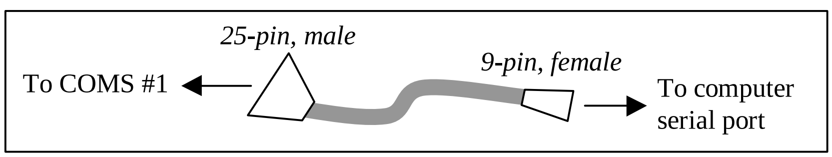

Figure 12. Computer data cable assembly.

RS-232 25-pin/9-pin D-subminiature connector pin assignments at COMS #1:

Color	*Pin # in 25-pin male connector*	*Pin # in 9-pin female connector*
brown/white	8	1
white/brown	3	2
orange/white	2	3
white/orange	20	4
green/white	7	5
white/green	6	6
gray/white	4	7
white/gray	5	8
blue/white	22	9

Appendix E. Code-Operated Matrix Switches (COMS)

The code-operated matrix switches are used to connect the thermometer, barometer and hygrometer to each station as needed. The COMS allows each of its ports to connect and disconnect to each other as needed using commands specific to the COMS.

Each port is configured using a "DIP" switch located inside each unit. The "DIP" switches are named using a SW*X* format where *X* is a letter. The switch names are listed below. Note that switches SWF and SWG are not used to configure a single port. Switch SWF is a system switch and configures all ports. The manufacturer reserves Switch SWG for future use.

Code-operated matrix switch #1: 9 ports

Port number	*Purpose*
0	connected to port 0 of COMS #2
1	computer for station #1
2	computer for station #2
3	computer for station #3
4	computer for station #4
5	balance for station #1
6	balance for station #2
7	balance for station #3
8	balance for station #4

Port settings:

Port number	*DIP switch designation*	*Switch configuration (1 = on, 0 = off)*
0	SWA	10000110
1	SWB	10000110
2	SWC	10000110
3	SWD	10000110
4	SWE	10000110
N/A	SWF	11010000
N/A	SWG	not used
5	SWH	11001010
6	SWI	11001010
7	SWJ	11001010
8	SWK	11001010

Code-operated matrix switch #2: 4 ports

Port number	*Purpose*
0	connected to port 0 of COMS #1
1	not used
2	barometer
3	hygrometer
4	thermometer

Port settings:

Port number	*DIP switch designation*	*Configuration (1 = on, 0 = off)*
0	SWA	10000110
1	SWB	10000110
2	SWC	00100110
3	SWD	11011110
4	SWE	10000110
N/A	SWF	10010000
N/A	SWG	not used

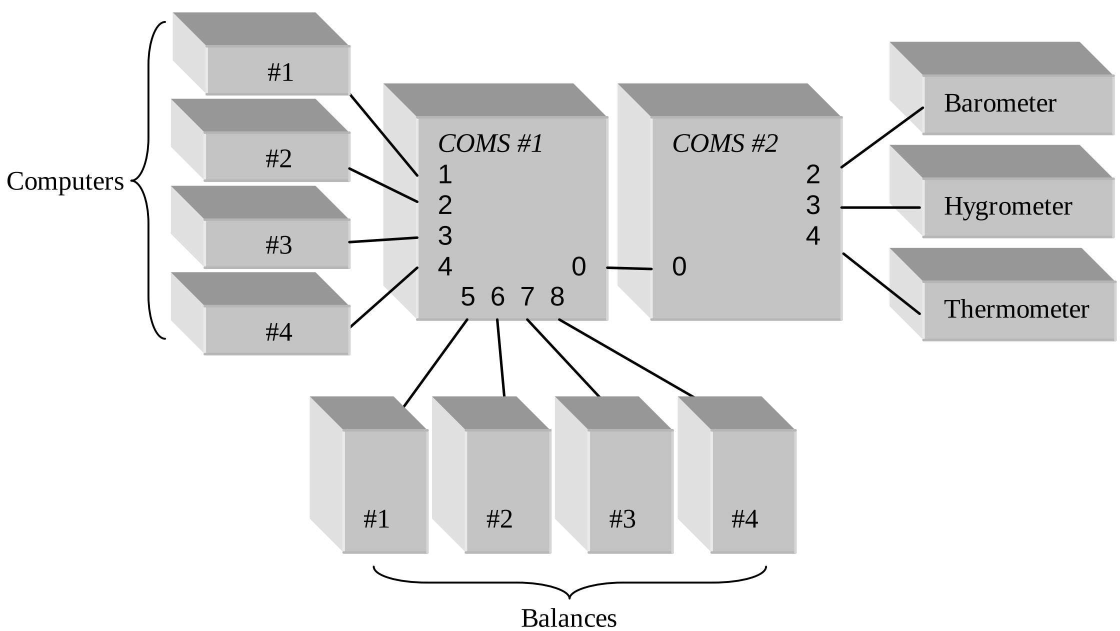

Figure 13. Equipment/COMS port connections.

Appendix F. Device Communication Protocol

The following are descriptions of the commands and status used by the automation software to communicate with the devices and code-operated matrix switches.

Code-operated matrix switch protocol

To link ports, the **L** command is used. The format is <*atten_char*>**L**<*port_num_a*><*port_num_b*> where: <*atten_char*>is the attention character, either **%** for COMS #1, or **@** for COMS #2 and <*port_num_a*> and <*port_num_b*>are single-digit numbers indicating which ports are to be connected together. An example would be **%L01**. This links ports #0 and #1 together in COMS #1 when they are available. The return status may be **Link established**.

To unlink the ports, the **U** command is used. The format is <*atten_char*>**U**<*port_num_a*><*port_num_b*> where <*atten_char*>is the attention character, **%** for COMS #1 or **@** for COMS #2 and <*port_num_a*>and <*port_num_b*> are single-digit numbers telling which ports are to be connected together. An example would be **@U01**. This unlinks port #0 from #1 in COMS #2. The return status may be **Link undone** or **Link does not exist**.

The query command is used to find what links exists in the COMS. This command format is <*atten_char*>**Q** where **Q** is the query command. A sample status string (with a link between ports 0 and 2) returned by the COMS looks like:

Port	Name	Linked To	Transparent	Activity	Attention String
0		..2....	N	N	%
1		....	N	N	%
2		0......	N	N	%
3		....	N	N	%
4		....	N	N	%
5		....	N	N	@
6		....	N	N	@
7		....	N	N	@
8		....	N	N	@

This string is parsed for the linkages.

Barometer protocol

To get the pressure from the barometer, the command ***0100P** is sent. A value is returned in the status format ***0001P=**<*pressure*> where <*pressure*> is the air pressure read in mm-Hg. A sample is ***0001P=762.0045**.

Before running the measurements the barometer is checked. There are two commands used. They are ***0100MC** (memory check) and ***0100CS** (check stack). The returned values for the memory check may be ***0001MC=Y** or ***0001MC=N**. The first indicates that the memory check went successfully and the second means an unsuccessful check. This returns the number of unused bytes in the microprocessor operations stack since the barometer was powered on. This number is not of significance for the application so the program only checks for a non-zero status message length. If not zero, the command is assumed to have returned a successful status.

After these two checks, initialization commands are sent. They are ***9900BR=1200** and ***9900PT=N**. The first sets the baud rate to 1200 and the second sets the parity to none, data bits to 8, and one stop bit.

Hygrometer protocol

The **S** command is sent to the hygrometer to initialize it. The greater-than character, **>**, is returned by the hygrometer. The **S** command allows the hygrometer to respond to commands it receives until a command is sent to stop responding. During the automation, the hygrometer is never made to stop responding to commands.

To read the hygrometer, the command **send** is sent. The relative humidity is returned along with other values such as temperature, dew point temperature, absolute humidity, mixing ratio, and wet bulb temperature. From this the relative humidity is parsed for and retrieved.

A sample of the hygrometer status message looks like:

```
    RH      T       Td      a       X       Tw   <cr><lf>
     39.5    23.2    8.7     8.2     7.0    14.8<cr><lf>
  *****.******.******.******.******.******.*<cr><lf>
```

where <*cr*><*lf*> are the carriage return and line feed characters that end each line. The message contains the readings from the probes attached to the hygrometer. The first line contains labels for the items measured. They are the relative humidity (RH) in percent, temperature (T) in ºC, dew-point temperature (Td) in ºC, absolute humidity (a) in g/m ^{3}, mixing ratio (X) in g/kg, and wet-bulb temperature (Tw) in ºC. the second line contains the measurements by the first probe, the third line, the second probe. Since the second probe is not installed, only asterisks appear in place of numbers.

No other commands are sent to the hygrometer for the duration of the measurement.

Thermometer protocol

The thermometer is initialized by two commands: **U0** and **R1**. **U0** sets the units to Celsius and **R1** sets the returned values to three decimal places.

To read the thermometer, the probe attached to the station must first be chosen. The **SA** command is used. The format for this command is **SA**<*channel_number*> where <*channel_number*>is a 2-digit number between 00 and 07 indicating the channel number in probe A. In this automation the numbers 01, 02, 03, 04 are used for station #1, #2, #3, and #4, respectively. These channels are connected to probe A. To choose the probe for station #1, the command **SA01** is sent to the thermometer.

After successfully choosing the probe, the command **MI** is sent for the temperature. The returned message has the format **A**<*temperature*>**C**<*channel_number*>. The **A** indicates the probe, <*temperature*>is the air temperature, **C** indicates the units, and <*channel_number*>is the channel chosen. If the returned message does not contain the correct values, the **MI** command is sent again. If it is correct, the air temperature is read by parsing.

The thermometer receives these commands from each of the workstations with parameters corresponding to the particular workstation. The thermometer responds with the reading from the particular probe chosen.

Balance protocol

To read the balance, the command **S** is used. This is a request for a stable reading. In return, there are three possible answers: **S** <*reading*>, **SI+**, or **SI-**. **S** <*reading*> is a stable reading where <*reading*> is the balance reading followed by the units. A sample would be **S 0.0001 mg**. **SI+** indicates an overload condition. **SI-** indicates an underload condition. If either of these values is found, the **S** command is sent again.

When opening or closing the balance the “beep” command is sent. This is an audible signal for the operator to take notice of the impending balance movement. The command is **DB 2**. The **2** indicates a double beep.

To open or close the balance door the **WI** command is used. The format for this command is **WI** <*option*> where <*option*>is either **0**, to open the door, or **1**, to close the door.

As a visual signal for the operator, the **D** command is used to display an abbreviated message on the balance display. The format is **D** <*text*> where <*text*>is the message to display. Messages displayed include **Opening…** which signals the opening of the balance door, **Closing…** which signals the closing of the balance door, and **Place…** which instructs the operator to place weights on the balance pan. Abbreviated messages are used since the balances’ displays are not very large. The complete message is displayed on the computer screen.

Appendix G. List of Design Matrix and Vector Files

The design matrix files have the extension *.mtx and the design vector files have the same name but the extension *.vct. They are stored in the directory c:\autobal\system. The matrix files are used to describe to the operator the design chosen and the vector files are used to control the measurement sequence. Refer to NBS Technical Note 952[4] for information on these designs.

Filename	*Design*
16a_i_41	4-1 for "5,2,2,1,1,1 series"
16a_i_51	5-1 for "5,2,2,1,1,1 series"
16a_ii	5,2,2,1,1,1
16a_iii	5,2,2,1,1,1
16a_iv	5,2,2,1,1,1
16a_v	5,2,2,1,1,1
16a_vi	5,2,2,1,1,1
16a_vii	5,2,2,1,1,1
31s	1,1,1
41s	1,1,1,1
51s	1,1,1,1,1
52j_i_41	4-1 for "5,3,2,1,1 series"
52j_i_51	5-1 for "5,3,2,1,1 series"
52j_ii	5,3,2,1,1
52j_iii	5,3,2,1,1
52j_iv	5,3,2,1,1
52j_v	5,3,2,1,1
52j_vi	5,3,2,1,1
52j_vii	5,3,2,1,1
R62c_i_41	4-1 for "5,3,2,1,1,1 series"
R62c_i_51	5-1 for "5,3,2,1,1,1 series"
R62c_ii	5,3,2,1,1,1
R62c_iii	5,3,2,1,1,1
R62c_iv	5,3,2,1,1,1
R62c_v	5,3,2,1,1,1
R62c_vi	5,3,2,1,1,1
R62c_vii	5,3,2,1,1,1
5321111_i	5,3,2,1,1,1,1
5321111_ii	5,3,2,1,1,1,1
5321111_iii	5,3,2,1,1,1,1
5321111_iv	5,3,2,1,1,1,1
5321111_v	5,3,2,1,1,1,1
5321111_vi	5,3,2,1,1,1,1
5321111_vii	5,3,2,1,1,1,1
5221111_i	5,2,2,1,1,1,1
5221111_ii	5,2,2,1,1,1,1
5221111_iii	5,2,2,1,1,1,1
5221111_iv	5,2,2,1,1,1,1
5221111_v	5,2,2,1,1,1,1
5221111_vi	5,2,2,1,1,1,1
5221111_vii	5,2,2,1,1,1,1

Note that for the set of series types 16A, 52J, and R62C, there are the "4-1 for …" and "5-1 for …" designs. Only one or the other is used as the first series in the set.

[4] Cameron, J. M.; Croarkin, M. C.; Raybold, R. C.; "Designs for the Calibration of Standards of Mass," NBS Technical Note 952; June 1977; U.S. Department of Commerce.

Appendix H. Contents of the Design Matrix Files

This appendix lists the contents of the design matrix files. After the end of the contents there is only an end-of-file character – no carriage return or linefeed exists there. Note that the files, 16a_i_41, 16a_i_51, 52j_i_41, 52j_i_51, R62c_i_41 and R62c_i_51 contain the 4-1 or 5-1 designs (as denoted by the last characters of the filename).

16a_i_41.mtx

```
+,-,0,0
+,0,-,0
+,0,0,-
0,+,-,0
0,+,0,-
0,0,+,-
```

16a_i_51.mtx

```
+,-,0,0,0
+,0,-,0,0
+,0,0,-,0
+,0,0,0,-
0,+,-,0,0
0,+,0,-,0
0,+,0,0,-
0,0,+,-,0
0,0,+,0,-
0,0,0,+,-
```

16a_ii.mtx,
16a_iii.mtx,
16a_iv.mtx,
16a_v.mtx,
16a_vi.mtx,
16a_vii.mtx

```
+,-,-,-,-,+
+,-,-,-,+,-
+,-,-,+,-,-
+,-,0,-,-,-
+,0,-,-,-,-
0,+,-,+,-,0
0,+,-,-,0,+
0,+,-,0,+,-
```

31s.mtx

```
+,-,0
+,0,-
0,+,-
```

41s.mtx

```
+,-,0,0
+,0,-,0
+,0,0,-
0,+,-,0
0,+,0,-
0,0,+,-
```

51s.mtx

```
+,-,0,0,0
+,0,-,0,0
+,0,0,-,0
+,0,0,0,-
0,+,-,0,0
0,+,0,-,0
0,+,0,0,-
0,0,+,-,0
0,0,+,0,-
0,0,0,+,-
```

52j_i_41.mtx

```
+,-,0,0
+,0,-,0
+,0,0,-
0,+,-,0
0,+,0,-
0,0,+,-
```

52j_i_51.mtx

```
+,-,0,0,0
+,0,-,0,0
+,0,0,-,0
+,0,0,0,-
0,+,-,0,0
0,+,0,-,0
0,0,+,0,-
0,0,+,-,0
0,0,+,0,-
0,0,0,+,-
```

52j_ii.mtx,
52j_iii.mtx,
52j_iv.mtx,
52j_v.mtx,
52j_vi.mtx,
52j_vii.mtx

```
+,-,-,+,-
+,-,-,-,+
+,-,-,0,0
+,-,0,-,-
0,+,-,-,0
0,+,-,0,-
0,0,+,-,-
0,0,0,+,-
```

R62c_i_41.mtx

```
+,-,0,0
+,0,-,0
+,0,0,-
0,+,-,0
0,+,0,-
0,0,+,-
```

R62c_i_51.mtx

```
+,-,0,0,0
+,0,-,0,0
+,0,0,-,0
+,0,0,0,-
0,+,-,0,0
0,+,0,-,0
0,+,0,0,-
0,0,+,-,0
0,0,+,0,-
0,0,0,+,-
```

R62c_ii.mtx,
R62c_iii.mtx,
R62c_iv.mtx,
R62c_v.mtx,
R62c_vi.mtx,
R62c_vii.mtx

```
+,-,-,+,-,0
+,-,-,0,+,-
+,-,-,-,0,+
+,-,-,0,0,0
+,0,-,-,-,-
0,+,-,+,-,-
0,+,-,-,+,-
0,+,-,-,-,+
0,0,+,-,-,0
0,0,+,-,0,-
0,0,+,0,-,-
```

5321111_i.mtx,
5321111_ii.mtx,
5321111_iii.mtx,
5321111_iv.mtx,
5321111_v.mtx,
5321111_vi.mtx,
5321111_vii.mtx

```
+,-,-,0,0,0,0
+,-,0,-,-,0,0
+,-,0,0,0,-,-
0,+,-,0,-,0,0
0,+,-,-,0,0,0
0,+,-,0,0,-,0
0,+,-,0,0,0,-
0,0,+,-,-,0,0
0,0,+,0,-,-,0
0,0,+,0,0,-,-
0,0,+,-,0,0,-
```

5221111_i.mtx,
5221111_ii.mtx,
5221111_iii.mtx,
5221111_iv.mtx,
5221111_v.mtx,
5221111_vi.mtx,
5221111_vii.mtx

```
+,-,-,-,0,0,0
+,-,-,0,-,0,0
+,-,-,0,0,-,0
+,-,-,0,0,0,-
0,+,0,+,-,-,-
0,0,+,+,-,-,-
0,+,+,-,-,-,-
0,0,0,0,+,-,0
0,0,0,0,+,0,-
0,0,0,0,0,+,0
```

Appendix I. Contents of Design Vector Files

This appendix lists the contents of the design vector files. After the end of the contents there is only an end-of-file character – no carriage return or linefeed exists there.

16a_i_41.vct [5]

```
+,0,0,0
0,-,0,0
0,-,0,0
+,0,0,0
+,0,0,0
0,0,-,0
0,0,-,0
+,0,0,0
+,0,0,0
0,0,0,-
0,0,0,-
+,0,0,0
0,+,0,0
0,0,-,0
0,0,-,0
0,+,0,0
0,+,0,0
0,0,0,-
0,0,0,-
0,+,0,0
0,0,+,0
0,0,0,-
0,0,0,-
0,0,+,0
```

16a_i_51.vct

```
+,0,0,0,0
0,-,0,0,0
0,-,0,0,0
+,0,0,0,0
+,0,0,0,0
0,0,-,0,0
0,0,-,0,0
+,0,0,0,0
+,0,0,0,0
0,0,0,-,0
0,0,0,-,0
+,0,0,0,0
+,0,0,0,0
0,0,0,0,-
0,0,0,0,-
+,0,0,0,0
0,+,0,0,0
0,0,-,0,0
0,0,-,0,0
0,+,0,0,0
0,+,0,0,0
0,0,0,-,0
0,0,0,-,0
0,+,0,0,0
0,+,0,0,0
0,0,0,0,-
0,0,0,0,-
0,+,0,0,0
0,0,+,0,0
0,0,0,-,0
0,0,0,-,0
0,0,+,0,0
0,0,+,0,0
0,0,0,0,-
0,0,0,0,-
0,0,+,0,0
0,0,0,+,0
0,0,0,0,-
0,0,0,0,-
0,0,0,+,0
```

16a_ii.vct,
16a_iii.vct,
16a_iv.vct,
16a_v.vct,
16a_vi.vct,
16a_vii.vct

```
+,0,0,0,0,+
0,-,-,-,-,0
0,-,-,-,-,0
+,0,0,0,0,+
+,0,0,0,+,0
0,-,-,-,0,-
0,-,-,-,0,-
+,0,0,0,+,0
+,0,0,+,0,0
0,-,-,0,-,-
0,-,-,0,-,-
+,0,0,+,0,0
+,0,0,0,0,0
0,-,0,-,-,-
0,-,0,-,-,-
+,0,0,0,0,0
0,0,-,-,-,-
0,0,-,-,-,-
+,0,0,0,0,0
+,0,0,0,0,0
0,+,0,+,0,0
0,0,-,0,-,0
0,0,-,0,-,0
0,+,0,+,0,0
0,+,0,0,0,+
0,0,-,-,0,0
0,0,-,-,0,0
0,+,0,0,0,+
0,+,0,0,+,0
0,0,-,0,0,-
0,0,-,0,0,-
0,+,0,0,+,0
```

[5] Note that 16a_i_41 and 16a_i_51 are the same as the 4-1 and 5-1 designs, respectively. Either one is used in the first series in a set of series using the 16a design.

31s.vct [6]

```
+,0,0
0,-,0
0,-,0
+,0,0
+,0,0
0,0,-
0,0,-
+,0,0
0,+,0
0,0,-
0,0,-
0,+,0
```

41s.vct

```
+,0,0,0
0,-,0,0
0,-,0,0
+,0,0,0
+,0,0,0
0,0,-,0
0,0,-,0
+,0,0,0
+,0,0,0
0,0,0,-
0,0,0,-
+,0,0,0
0,+,0,0
0,0,-,0
0,0,-,0
0,+,0,0
0,+,0,0
0,0,0,-
0,0,0,-
0,+,0,0
0,0,+,0
0,0,0,-
0,0,0,-
0,0,+,0
```

51s.vct

```
+,0,0,0,0
0,-,0,0,0
0,-,0,0,0
+,0,0,0,0
+,0,0,0,0
0,0,-,0,0
0,0,-,0,0
+,0,0,0,0
+,0,0,0,0
0,0,0,-,0
0,0,0,-,0
+,0,0,0,0
+,0,0,0,0
0,0,0,0,-
0,0,0,0,-
+,0,0,0,0
0,+,0,0,0
0,0,-,0,0
0,0,-,0,0
0,+,0,0,0
0,+,0,0,0
0,0,0,-,0
0,0,0,-,0
0,+,0,0,0
0,+,0,0,0
0,0,0,0,-
0,0,0,0,-
0,+,0,0,0
0,0,+,0,0
0,0,0,-,0
0,0,0,-,0
0,0,+,0,0
0,0,+,0,0
0,0,0,0,-
0,0,0,0,-
0,0,+,0,0
0,0,0,+,0
0,0,0,0,-
0,0,0,0,-
0,0,0,+,0
```

[6] 31S, 41S and 51S are stand-alone series and are not used as the first series in a set of series.

52j_i_41.vct [7]

```
+,0,0,0
0,-,0,0
0,-,0,0
+,0,0,0
+,0,0,0
0,0,-,0
0,0,-,0
+,0,0,0
+,0,0,0
0,0,0,-
0,0,0,-
+,0,0,0
0,+,0,0
0,0,-,0
0,0,-,0
0,+,0,0
0,+,0,0
0,0,0,-
0,0,0,-
0,+,0,0
0,0,+,0
0,0,0,-
0,0,0,-
0,0,+,0
```

52j_i_51.vct

```
+,0,0,0,0
0,-,0,0,0
0,-,0,0,0
+,0,0,0,0
+,0,0,0,0
0,0,-,0,0
0,0,-,0,0
+,0,0,0,0
+,0,0,0,0
0,0,0,-,0
0,0,0,-,0
+,0,0,0,0
+,0,0,0,0
0,0,0,0,-
0,0,0,0,-
+,0,0,0,0
0,+,0,0,0
0,0,-,0,0
0,0,-,0,0
0,+,0,0,0
0,+,0,0,0
0,0,0,-,0
0,0,0,-,0
0,+,0,0,0
0,+,0,0,0
0,0,0,0,-
0,0,0,0,-
0,+,0,0,0
0,0,+,0,0
0,0,0,-,0
0,0,0,-,0
0,0,+,0,0
0,0,+,0,0
0,0,0,0,-
0,0,0,0,-
0,0,+,0,0
0,0,0,+,0
0,0,0,0,-
0,0,0,0,-
0,0,0,+,0
```

52j_ii.vct,
52j_iii.vct,
52j_iv.vct,
52j_v.vct,
52j_vii.vct,
52j_viii.vct

```
+,0,0,+,0
0,-,-,0,-
0,-,-,0,-
+,0,0,+,0
+,0,0,0,+
0,-,-,-,0
0,-,-,-,0
+,0,0,0,+
+,0,0,0,0
0,-,-,0,0
0,-,-,0,0
+,0,0,0,0
+,0,0,0,0
0,-,0,-,-
0,-,0,-,-
+,0,0,0,0
0,+,0,0,0
0,0,-,-,0
0,0,-,-,0
0,+,0,0,0
0,+,0,0,0
0,0,-,0,-
0,0,-,0,-
0,+,0,0,0
0,0,+,0,0
0,0,0,-,-
0,0,0,-,-
0,0,+,0,0
0,0,0,+,0
0,0,0,0,-
0,0,0,0,-
0,0,0,+,0
```

[7] Note that 52j_i_41 and 52j_i_51 are the same as the 4-1 and 5-1 designs, respectively. Either one is used in the first series in a set of series using the 16a design.

R62c_i_41.vct [8]

```
+,0,0,0
0,-,0,0
0,-,0,0
+,0,0,0
+,0,0,0
0,0,-,0
0,0,-,0
+,0,0,0
+,0,0,0
0,0,0,-
0,0,0,-
+,0,0,0
0,+,0,0
0,0,-,0
0,0,-,0
0,+,0,0
0,+,0,0
0,0,0,-
0,0,0,-
0,+,0,0
0,0,+,0
0,0,0,-
0,0,0,-
0,0,+,0
```

R62c_i_51.vct

```
+,0,0,0,0
0,-,0,0,0
0,-,0,0,0
+,0,0,0,0
+,0,0,0,0
0,0,-,0,0
0,0,-,0,0
+,0,0,0,0
+,0,0,0,0
0,0,0,-,0
0,0,0,-,0
+,0,0,0,0
+,0,0,0,0
0,0,0,0,-
0,0,0,0,-
+,0,0,0,0
0,+,0,0,0
0,0,-,0,0
0,0,-,0,0
0,+,0,0,0
0,+,0,0,0
0,0,0,-,0
0,0,0,-,0
0,+,0,0,0
0,+,0,0,0
0,0,0,0,-
0,0,0,0,-
0,+,0,0,0
0,0,+,0,0
0,0,0,-,0
0,0,0,-,0
0,0,+,0,0
0,0,+,0,0
0,0,0,0,-
0,0,0,0,-
0,0,+,0,0
0,0,0,+,0
0,0,0,0,-
0,0,0,0,-
0,0,0,+,0
```

R62c_ii.vct,
R62c_iii.vct,
R62c_iv.vct,
R62c_v.vct,
R62c_vi.vct,
R62c_vii.vct

```
+,0,0,+,0,0
0,-,-,0,-,0
0,-,-,0,-,0
+,0,0,+,0,0
+,0,0,0,+,0
0,-,-,0,0,-
0,-,-,0,0,-
+,0,0,0,+,0
+,0,0,0,0,+
0,-,-,-,0,0
0,-,-,-,0,0
+,0,0,0,0,+
+,0,0,0,0,0
0,-,-,0,0,0
0,-,-,0,0,0
+,0,0,0,0,0
+,0,0,0,0,0
0,0,-,-,-,-
0,0,-,-,-,-
+,0,0,0,0,0
0,+,0,+,0,0
0,0,-,0,-,-
0,0,-,0,-,-
0,+,0,+,0,0
0,+,0,0,+,0
0,0,-,-,0,-
0,0,-,-,0,-
0,+,0,0,+,0
0,+,0,0,0,+
0,0,-,-,-,0
0,0,-,-,-,0
0,+,0,0,0,+
0,0,+,0,0,0
0,0,0,-,-,0
0,0,0,-,-,0
0,0,+,0,0,0
0,0,+,0,0,0
0,0,0,-,0,-
0,0,0,-,0,-
0,0,+,0,0,0
0,0,+,0,0,0
0,0,0,0,-,-
0,0,0,0,-,-
0,0,+,0,0,0
```

[8] Note that R62c_i_41 and R62c_i_51 are the same as the 4-1 and 5-1 designs, respectively. Either one is used in the first series in a set of series using the 16a design.

5221111_i.vct,
5221111_ii.vct,
5221111_iii.vct,
5221111_iv.vct,
5221111_v.vct,
5221111_vi.vct,
5221111_vii.vct

```
+,0,0,0,0,0,0
0,-,-,-,0,0,0
0,-,-,-,0,0,0
+,0,0,0,0,0,0
+,0,0,0,0,0,0
0,-,-,0,-,0,0
0,-,-,0,-,0,0
+,0,0,0,0,0,0
+,0,0,0,0,0,0
0,-,-,0,0,-,0
0,-,-,0,0,-,0
+,0,0,0,0,0,0
+,0,0,0,0,0,0
0,-,-,0,0,0,-
0,-,-,0,0,0,-
+,0,0,0,0,0,0
0,+,0,+,0,0,0
0,0,0,0,-,-,-
0,0,0,0,-,-,-
0,+,0,+,0,0,0
0,0,+,+,0,0,0
0,0,0,0,-,-,-
0,0,0,0,-,-,-
0,0,+,+,0,0,0
0,+,+,0,0,0,0
0,0,0,-,-,-,-
0,0,0,-,-,-,-
0,+,+,0,0,0,0
0,0,0,0,+,0,0
0,0,0,0,0,-,0
0,0,0,0,0,-,0
0,0,0,0,+,0,0
0,0,0,0,+,0,0
0,0,0,0,0,0,-
0,0,0,0,0,0,-
0,0,0,0,+,0,0
0,0,0,0,0,+,0
0,0,0,0,0,0,-
0,0,0,0,0,0,-
0,0,0,0,0,+,0
```

5321111_i.vct,
5321111_ii.vct,
5321111_iii.vct,
5321111_iv.vct,
5321111_v.vct,
5321111_vi.vct,
5321111_vii.vct

```
+,0,0,0,0,0,0
0,-,-,0,0,0,0
0,-,-,0,0,0,0
+,0,0,0,0,0,0
+,0,0,0,0,0,0
0,-,0,-,-,0,0
0,-,0,-,-,0,0
+,0,0,0,0,0,0
+,0,0,0,0,0,0
0,-,0,0,0,-,-
0,-,0,0,0,-,-
+,0,0,0,0,0,0
0,+,0,0,0,0,0
0,0,-,0,-,0,0
0,0,-,0,-,0,0
0,+,0,0,0,0,0
0,+,0,0,0,0,0
0,0,-,-,0,0,0
0,0,-,-,0,0,0
0,+,0,0,0,0,0
0,+,0,0,0,0,0
0,0,-,0,0,-,0
0,0,-,0,0,-,0
0,+,0,0,0,0,0
0,+,0,0,0,0,0
0,0,-,0,0,0,-
0,0,-,0,0,0,-
0,+,0,0,0,0,0
0,0,+,0,0,0,0
0,0,0,-,-,0,0
0,0,0,-,-,0,0
0,0,+,0,0,0,0
0,0,+,0,0,0,0
0,0,0,0,-,-,0
0,0,0,0,-,-,0
0,0,+,0,0,0,0
0,0,+,0,0,0,0
0,0,0,0,0,-,-
0,0,0,0,0,-,-
0,0,+,0,0,0,0
0,0,+,0,0,0,0
0,0,0,-,0,0,-
0,0,0,-,0,0,-
0,0,+,0,0,0,0
```

Appendix J. Equipment Power Usage

The power usage of each of the instruments included in the automation is listed in the table below. Note that the UMT-type balances are not currently part of the automation but are connected to the same uninterrupted power supply in the laboratory.

Thermometer and connector box	42 W
COMS #1	45 W
COMS #2	45 W
Barometer	12 W
Hygrometer	5.4 W
AT21 balance	15.5 W
AT106 balance	15.5 W
AT20 balance	15.5 W
AT1005 balance	15.5 W
UMT5 balance	15.5 W
UMT5 balance	15.5 W
UMT2 balance	15.5 W

Appendix K. List of Instrument Operator Manuals

1. Code Operated Matrix Switch and 4-Port Expansion Board, December 1992, SW540A-R3, SW541, ©1992, Black Box Corporation

2. Digiquartz® Precision Pressure Instruments, Document No. 8107-001, Revision J, July 1995, ©1995, Paroscientific, Inc., Redmond, Washington.

3. HMI38 Humidity Data Processor, Operating Manual, HMI38-U064en-1.3, 16 January, 1996, ©1996, Vaisala, Woburn, Massachusetts.

4. Model 250 Precision Thermometer, Operator's Handbook, Automatic Systems Laboratories, Inc., Wilmington, Massachusetts.

5. Operating Instructions; Bi-directional Data Interface; Mettler AT/MT/UMT Balances, ME-7040198A, ©1990, Mettler-Toledo AG, Greifensee, Switzerland.